

JavaScript

Prof. Daniel Di Domenico

daniel.domenico@ifpr.edu.br

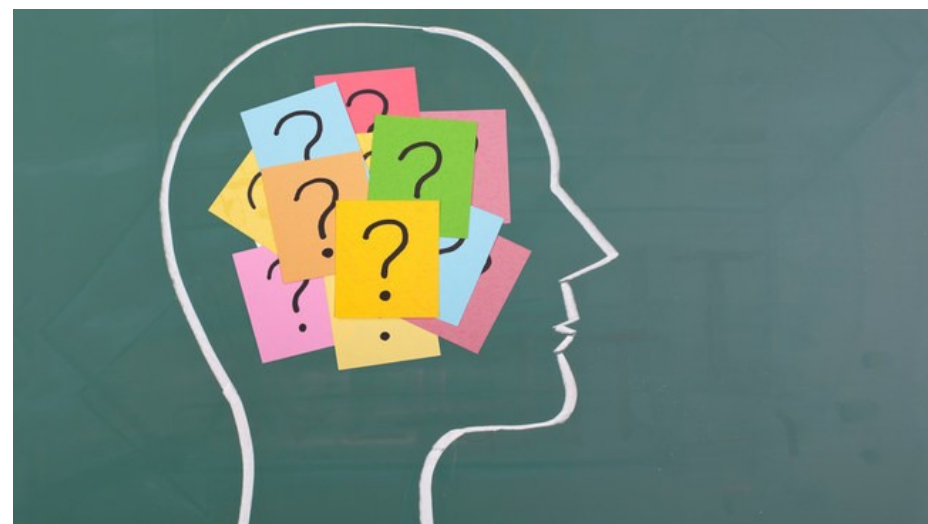


Introdução

Revisão...

- **O que já sabemos:**

- HTML para criar o conteúdo das páginas Web
- CSS para adicionar estilos às páginas Web
 - Cores, tamanhos, posicionamento, bordas, efeitos...
 - Seletores: elemento, ID, classe, pseudo-classe e pseudo-elemento



O que é JavaScript?

- JavaScript (JS):
 - **Linguagem de programação**
 - Linguagem de script e interpretada
 - Também conhecida como linguagem de script para páginas Web
 - É suportada por todos os navegadores modernos (Chrome, Firefox, Safari...)
 - Comandos são executados pelo navegador
 - **ATENÇÃO:** não confundir com Java
 - Ambas são linguagens de programação, mas possuem sintaxe, semânticas e usos muito diferentes

Por que utilizar JavaScript?

- JS é uma linguagem largamente utilizada no mundo
 - Forma a **tríade do desenvolvimento Web** junto com HTML e CSS
 - Padrão W3C (World Wide Web Consortium)
 - Tutorial oficial do JS: <https://www.w3schools.com/js/>
- Permite a implementação de **elementos dinâmicos** em páginas Web
 - Utilizando apenas HTML e CSS, as páginas são estáticas
 - JS possibilita manipular, calcular e validar informações
 - Atualização e aperfeiçoamento dos códigos HTML e CSS



Funcionalidades do JS

- **Detectar eventos** (interações do usuário) que ocorrem em uma página Web, como:
 - Clique em um elemento (botão ou link)
 - Digitação em um campo texto
 - Seleção de uma opção de seleção
 - Foco em um elemento
- **Responder aos eventos** através de ações, como:
 - Exibir uma mensagem
 - Mudar o conteúdo de um elemento
 - Alterar o estilo dos elementos
 - Validar dados
 - Exibir e ocultar elementos

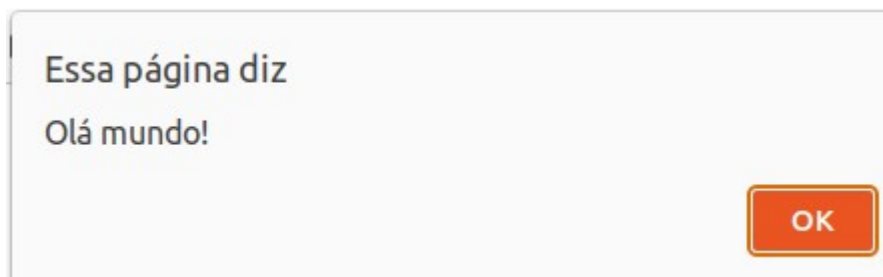
Características do JS

- Código fonte JS é incluído junto ao HTML
 - Pode ser feito de diferentes formas
- Programação é dirigida por eventos e orientada a objetos
- Tipagem do JS
 - Fraca: pode-se alterar o tipo de uma variável
 - Dinâmica: a variável assume o tipo do valor atribuído
- JS é **case-sensitive**
 - **Diferencia letras maiúsculas de minúsculas**
 - Utilize o padrão camelCase

Primeiro exemplo com JS

- Exibir mensagem com JS

```
<button type="button" onclick="alert('Olá mundo!');">Clique aqui!</button>
```



Formas de exibir dados

- Com JS, podemos exibir dados de diferentes formas:

<!--Utilizando uma caixa de alerta-->

```
<button type="button" onclick="alert('Olá mundo!');">Alerta</button>
```

<!--Escrevendo no HTML-->

```
<button type="button" onclick="document.write('Olá mundo!');">Html</button>
```

<!--Escrevendo em um elemento da página HTML-->

```
<p id="paragrafo"></p>
```

```
<button type="button"  
  onclick="document.getElementById('paragrafo').innerHTML = 'Olá mundo!';">  
  Elemento Html</button>
```

<!--Escrevendo no console do navegador-->

```
<button type="button" onclick="console.log('Olá mundo!');">Console</button>
```


Adicionar JS na página HTML

- Forma 1:

- Junto ao HTML usando a tag **<script>**
- **<script type="text/javascript">codigo JS</script>**
 - Deve ser declarado antes do fim da tag **<body>** (melhor desempenho)
 - Utilize funções para poder executar o código de maneira controlada

type é opcional

```
<p id="paragrafo">Este texto vai ser alterado</p>
<button type="button" onclick="evento();">Executa Script</button>

<script>
  function evento() {
    document.getElementById('paragrafo').innerHTML = 'A mágica aconteceu!';
    document.getElementById('paragrafo').style.color = 'blue';
  }
</script>
```

Adicionar JS na página HTML

- Forma 2:
 - Scripts externos
 - Arquivo externo (.js), sendo adicionado através da tag **<script>**
 - Também deve ser declarado antes do fim da tag <body> (melhor desempenho)

```
<p id="paragrafo">Este texto vai ser alterado</p>
<button type="button" onclick="evento();">Executar script</button>

<script src="ola_mundo.js"></script>
```

```
//Arquivo ola_mundo.js
function evento() {
    document.getElementById('paragrafo').innerHTML = 'Abracadabra!';
    document.getElementById('paragrafo').style.color = 'green';
}
```

Variáveis

- Utilizar a palavra **var** para declarar as variáveis
- A tipagem é dinâmica funcionando de acordo com o valor atribuído

```
var x = 3; //variável int  
var y = "algoritmo"; //variável string  
var z = 1.8; //variável float  
var w = true; //variável boolean
```

```
//Checagem de tipo  
console.log(typeof(y)); //Imprime string
```

Operadores matemáticos

- Devem ser utilizados com o tipos numéricos

```
var x = 8;
```

```
var y = 5;
```

```
var a = x + y; //Resultado: 13
```

```
var b = x - y; //Resultado: 3
```

```
var c = x * y; //Resultado: 40
```

```
var d = x / y; //Resultado: 1.6
```

```
var d = x % y; //Resultado: 3 (operador de resto da divisão)
```

Concatenação de string

- É utilizada para concatenar strings a outros dados

```
var x = "texto";  
var y = "palavra";  
var z = 5;
```

```
var a = x + y; //Resultado: textopalavra  
var b = x + y; //Resultado: texto palavra  
var c = x + z; //Resultado: texto5
```

Operadores relacionais

- Comparações que resultam em Verdadeiro (true) ou Falso (false)

```
var x = 8;
```

```
var y = 5;
```

```
var a = x > y; //Resultado: true
```

```
var b = x < y; //Resultado: false
```

```
var c = x == y; //Igualdade – Resultado: false
```

```
var d = x != y; //Diferente - Resultado: true
```

```
var e = x >= y; //Resultado: true
```

```
var f = x <= y; //Resultado: false
```

Operadores relacionais

- Comparações que resultam em Verdadeiro (true) ou Falso (false)

```
var x = true;
```

```
var y = false;
```

```
var a = x && y; //E – Resultado: false
```

```
var b = x || y; //OU - Resultado: true
```

```
var c = ! x; //NOT (NÃO) – Resultado: false
```

Converter string para número

- É necessário para fazer cálculos entre números

```
var x = "1" + 2;  
//Resultado: 12
```

```
var y = parseInt("1") + 2;  
//Resultado: 3
```

```
/* parseFloat converte para o tipo float
```


Funções

- Sub-rotinas para executar comandos específicos
 - **Vantagem:** reutilização de código
 - Podem ou não possuir parâmetros
 - Podem ou não retornar valores

Funções

- Função **sem parâmetros**:

```
function cotacaoDolar() {  
    return 5.35;  
}
```

```
//Chamada  
var valor = cotacaoDolar();
```

Funções

- **Função com parâmetros:**

```
function media(nota1, nota2, nota3){  
    var m = (nota1+nota2+nota3) / 3;  
    return m;  
}
```

//Chamada

```
var media = media(10, 8, 7.5);
```

Funções

- Função **sem retorno**:

```
function mostrarMensagem(texto){  
    alert(texto);  
}
```

//Chamada

```
mostrarMensagem("Mensagem a ser exibida!");
```

Exercícios

- **1-** Crie um página que chame uma função JavaScript através de um botão, sendo:
 - **soma(v1, v2, v3)**: deve receber 3 valores por parâmetro e realizar a soma dos mesmos, retornando o resultado;
- **2-** Crie outro botão na página, chamando a seguinte função
 - **exibeMsg(id, valor)**: deve receber dois parâmetros para exibir uma mensagem em um componente existente na pagina HTML:
 - Parâmetro 1: ID do elemento HTML
 - Parâmetro 2: valor a ser exibido no documento
 - [Dica: para exibir o resultado no elemento HTML, veja o slide 8](#)
- **3-** Crie um terceiro botão, chamando uma função JS que:
 - **3.1-** Chame a função soma() do exercício 1
 - **3.2-** Chame a função exibeMsg() do exercício 2 para exibir o resultado da soma